AF/1641 IFW

Docket No.: 057454-0060 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 20277
Isao MINEMATSU	:	Confirmation Number: 3710
Application No.: 09/819,990	:	Tech Center Art Unit: 2183
Filed: March 29, 2001	:	Examiner: Daniel H. Pan

For: MICROPROCESSOR EXECUTING DATA TRANSFER BETWEEN MEMORY AND REGISTER AND DATA TRANSFER BETWEEN REGISTERS IN RESPONSE TO SINGLE PUSH/POP INSTRUCTION

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

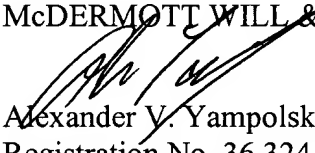
Submitted herewith is Appellant's Appeal Brief in support of the Notice of Appeal filed April 21, 2005. Please charge the Appeal Brief fee of $500.00 to Deposit Account 500417.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due under 37 C.F.R. 1.17 and 41.20, and in

connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP



Alexander V. Yampolsky
Registration No. 36,324

Please recognize our Customer No. 20277 as our correspondence address.

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 SAB/AVY/dlb
Facsimile: 202.756.8087
Date: June 15, 2005

Docket No.: 057454-0060 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 20277
Isao MINEMATSU	:	Confirmation Number: 3710
Application No.: 09/819,990	:	Tech Center Art Unit: 2183
Filed: March 29, 2001	:	Examiner: Daniel H. Pan

For: MICROPROCESSOR EXECUTING DATA TRANSFER BETWEEN MEMORY AND REGISTER AND DATA TRANSFER BETWEEN REGISTERS IN RESPONSE TO SINGLE PUSH/POP INSTRUCTION

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed April 21, 2005, wherein Appellant appeals from the Primary Examiner's rejection of claims 1-10.

Real Party In Interest

The real party in interest is Renesas Technology Corporation, the assignee of the entire right, title and interest in and to the above-identified U. S. Application.

Related Appeals and Interferences

No other appeals or interferences are known to the Appellant, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-10 are pending. Claims 11-14 are cancelled. Claims 1-10 stand under final rejection, from which rejection this appeal is taken.

Status of Amendments

According to the Advisory Action mailed April 11, 2005, the Amendment filed on March 21, 2005 will be entered for purposes of appeal.

Summary of Claimed Subject Matter

The claimed subject matter relates to a microprocessor capable of transferring data stored in multiple registers using a program having a reduced number of steps. As shown in FIG. 2 and described on pages 7-10, the microprocessor comprises an instruction decode unit 39 decoding an instruction fetched from an instruction memory 43, a program control unit (PCU) 40 controlling the fetch of instructions stored in the instruction memory 43, an address arithmetic unit (AAU) 41 producing addresses for accessing an X memory 44 or a Y memory 45 that store data, and a data arithmetic unit (DAU) 42 performing operations with data. In particular, the DAU 42 includes a multiplier 49, an arithmetic and logic unit (ALU) 50 and a shifter 51.

The microprocessor includes multiple registers illustrated in FIGS. 1A and 1B. In particular, the PCU 40 includes a register group 60 (FIG. 2) of 13 registers PC, PSW, BPC, BPSW, DPC, DPSW, PCLINK, LP_CT, REP_CT, LP_S, LP_E, PC_BRK and INT_S. The AAU 41 includes a register group 61 of 13 registers AR0 to AR3, AMD0 to AMD3,

AR_SEL, MOD_S, MOD_E, SP and AR_PAGE. The DAU 42 includes a register group 62 of four registers TR0 to TR3, a register group 63 of four registers R0 to R3 and a register group 64 of two registers A0 and A1. As described on pages 5-7, different types of registers are provided. For example, four work registers TR0 to TR3 temporarily hold addresses or data, four address registers AR0 to AR3 store addresses for memory access, four operation source registers R0 to R3 store operation source data.

The microprocessor further includes data buses D1-D6 for transferring data between the registers. Data buses 53, 54 and 55 are provided for transferring data between the registers and the X memory 44 or the Y memory 45.

The DAU 42 performs operations based on control signal D (reference number 48 in FIG. 2). For example, the control signal D may instruct the DAU 42 to perform reading from the X memory 44 or Y memory 45, or perform operations with content of the registers and writing results into the X memory 44 or the Y memory 45.

As described on pages 10-16 and illustrated in FIGS. 4-11, the DAU 42 executes data transfer between the registers, and data transfer between the registers and the memory 44 or 45 in accordance with a single instruction having a single operation code fetched by the PCU 40. In particular, data from a control register to a work register and from the work register to the X memory 44 are transferred based on a single push instruction, and data from the X memory 44 to a work register and from the work register to a control register are transferred based on a single pop instruction.

In particular, FIGS. 9A and 9B illustrate exemplary POP and PUSH instructions. For example, when the POP instruction (2) in FIG. 9A specifies a register, the value of the work register TR0 is transferred to the register specified by the POP instruction, data stored at the address of the X memory 44 specified by the stack pointer is transferred to the work register TR0 through the data buses 53 and 54, and thereafter the value of the stack pointer is incremented. Further, when the PUSH instruction (2) in FIG. 9B specifies a register, data stored in the work register TR0 is loaded at the address of the X memory 44 specified by the stack pointer, the value of the register specified by the PUSH instruction is transferred to the work register TR0, and thereafter the value of the stack pointer is decremented.

FIG. 10 illustrates an exemplary program for pushing data stored in the operation source register R0 and the address register AR0 and thereafter popping the data. For example, "push R0" shown as instruction (2) in FIG. 10 indicates that the value of the work register TR0 is loaded at the address of the X memory 44 indicated by the stack pointer, the value of the operation source register R0 is transferred to the work register TR0 and thereafter the value of the stack pointer is decremented. "Push AR0" shown as instruction (3) in FIG. 10 indicates that the value of the work register TR0 is loaded at the address of the X memory 44 indicated by the stack pointer, the value of the address register AR0 is transferred to the work register TR0 and thereafter the value of the stack pointer is decremented. "Pop AR0" shown as instruction (6) in FIG. 10 indicates that the value of the work register TR0 is transferred to the address register AR0, data stored at the address of the X memory 44 indicated by the stack pointer is transferred to the work register TR0 and thereafter the value of the stack pointer is incremented. "Pop R0" shown as instruction (7) in FIG. 10 indicates that the value of the work register TR0 is transferred to the operation source register R0, data stored at the

address of the X memory 44 indicated by the stack pointer is transferred to the work register TR0 and thereafter the value of the stack pointer is incremented.

Hence, in accordance with a claimed invention, a single instruction having a single operation code is sufficient to carry out data transfer between different registers as well as data transfer between the registers and the memory. As a result, the microprocessor is capable of performing operations with data held in multiple registers using a program having a reduced number of instructions.

Grounds of Rejection To Be Reviewed By Appeal

Whether claims 1-10 are unpatentable over Kudo et al. (6,560,692) in view of Geldman et al. (5,524,268) under 35 U.S.C. § 103(a).

Argument

In the application of a rejection under 35 U.S.C. §103, it is incumbent upon the Examiner to factually support a conclusion of obviousness. As stated in ***Graham v. John Deere Co.*** 383 U.S. 1, 13, 148 U.S.P.Q. 459, 465 (1966), obviousness under 35 U.S.C. §103 must be determined by considering (1) the scope and content of the prior art; (2) ascertaining the differences between the prior art and the claims in issue; and (3) resolving the level of ordinary skill in the pertinent art.

The Examiner must provide reasons why one having ordinary skill in the art would have been led to modify the prior art or to combine prior art references to arrive at the claimed invention. ***Ashland Oil, Inc. v. Delta Resins & Refractories, Inc.***, 776 F.2d 281, 227 USPQ 657 (Fed. Cir. 1985). ***In re Fine***, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir.

1988); ***Stratoflex, Inc. v. Aeroquip Corp.***, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983); ***In re Warner***, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967). These showings by the Examiner are an essential part of complying with the burden of presenting a *prima facie* case of obviousness. ***In re Oetiker***, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

As demonstrated below, the Examiner has failed to ascertain the differences between the prior art and the claims in issue, and provide the requisite reasons for modifying the references and thus to establish a *prima facie* case of obviousness..

In particular, claim 1 recites a microprocessor including:

- a program control unit controlling fetch of an instruction code;

- an instruction decode unit decoding said fetched instruction code;

- an address operation unit operating an address of a memory on the basis of the result of decoding by said instruction decode unit; and

- a data operation unit operating data on the basis of the result of decoding by said instruction decode unit.

The claim specifies that the data operation unit executes data transfer between registers and data transfer between the registers and the memory in correspondence to single said instruction code having a single operation code fetched by the program control unit.

Hence, claim 1 requires the data operation unit to execute data transfer between registers **as well as** data transfer between the registers and the memory in correspondence to **a single instruction code having a single operation code.**

The Examiner contends that Kudo differs from the claimed arrangement in that the reference does not disclose the data transfer between registers in correspondence to a single instruction. Geldman is relied upon for disclosing data transfer between registers in accordance with a single instruction MV8.

The Examiner takes the position that it would have been obvious to "use Geldman in Kudo for including data transfers between eh registers (sic) in correspondence to a single instruction as claimed because the use of Geldman can provide Kudo the processing capability to adapt to specific data transfer operations from different sources, such as between the registers in addition to the data transfer from the memory (sic)."

Considering the references, Kudo discloses a push instruction "pushn" provided to save values of registers R0-R3 in the stack (col. 15, lines 22-25).

Geldman discloses instruction MV8 for data transfer between registers (col. 8, lines 7-19). However, the reference specifically indicates that the instruction MV8 is composed of multiple operation codes (see table in col. 8, lines 10-15).

First, it is respectfully submitted that one skilled in the art would realize that Kudo does not need to transfer data between registers before saving the data from a register to the stack memory. Therefore, no reason is apparent to support the conclusion that one having ordinary skill in the art would have been impelled to add the data transfer between registers to Kudo operations performed in response to the push instruction. It is not apparent why one skilled in the art would have recognized any advantage to be gained by the proposed modification of Kudo in view of Geldman.

Further, even if Kudo were modified in view of the Geldman teaching, the claimed invention would not result.

In particular, neither Kudo nor Geldman suggests two data transfer operations, one of which is data transfer between registers and another - is data transfer between the registers and the memory, in correspondence to a single instruction code having a single operation code.

Therefore, if Kudo were modified in view of Geldman, the combined teachings of the Kudo and Geldman references would suggest performing a data transfer operation between registers in accordance with one instruction, and performing a data transfer operation between the registers and the memory in accordance with another instruction.

It is well settled that the test for obviousness is what the combined teachings of the references would have suggested to those having ordinary skill in the art. ***Cable Electric Products, Inc. v. Genmark, Inc.***, 770 F.2d 1015, 226 USPQ 881 (Fed. Cir. 1985). In determining whether a case of prima facie obviousness exists, it is necessary to ascertain whether the prior art teachings appear to be sufficient to one of ordinary skill in the art to suggest making the claimed substitution or other modification. ***In re Lalu***, 747 F.2d 703, 705, 223 USPQ 1257, 1258 (Fed. Cir. 1984).

As demonstrated above, the combined teachings of Kudo and Geldman are not sufficient to arrive at the claimed invention.

The Examiner has apparently failed to give adequate consideration to the particular problems and solution addressed by the claimed invention. ***Northern Telecom, Inc. v. Datapoint Corp.***, 908 F.2d 931, 15 USPQ2d 1321 (Fed. Cir. 1990); ***In re Rothermel***, 276 F.2d 393, 125 USPQ 328 (CCPA 1960). Specifically, as discussed above, the claimed invention suggests performing a data transfer between registers and a data transfer between the registers and a memory in accordance with a single instruction to simplify the program for controlling data transfer operations.

Neither Kudo nor Geldman addresses the problem and solution addressed by the claimed invention.

Moreover, as discussed above, the Geldman's instruction for data transfer between registers is composed of multiple operation codes. Therefore, the combined teachings would not teach the data transfer between registers in correspondence with a single instruction having a single operation code, as claim 1 requires.

It is noted that in the Advisory Action, the Examiner contends that "applicant never defines what is a "single operation code." The Examiner's position is respectfully traversed.

It is submitted that one skilled in the art armed with the specification and drawings would understand that the claimed instruction code having a single operation code is an instruction code composed of only one operation code.

Accordingly, Appellant submits that the lack of any motivation for the proposed combination of references to arrive at the claimed invention, coupled with the particular problems addressed and solved by the claimed invention, undermine the basis for the Examiner's rejection of claim 1 under 35 U.S.C. § 103.

Further, the applied reference combination would not suggest the subject matter of claims 2-9 dependent from claim 1.

In particular, the references do not teach or suggest that:

-the data operation unit transfers data stored in a first register to said memory and transfers data stored in a second register to said first register in correspondence to a single push instruction fetched by said program control unit, as claim 2 requires;

- the data operation unit decrements the value of a stack pointer after transferring the data stored in said second register to said first register, as claim 3 recites;

-the first register is a work register implemented in said data operation unit, as claims 4 and 8 recite;

-the second register is a control register implemented in one of said address operation unit and said program control unit, as claim 5 and 9 recite;

-the data operation unit transfers data stored in a first register to a second register and transfers data stored in said memory to said first register in correspondence to a single pop instruction fetched by said program control unit, as claim 6 recites;

-the data operation unit increments the value of a stack pointer after transferring said data stored in said memory to said first register, as claim 7 recites;

- the data operation unit transfers data stored in a first register to said memory and keeps the value of a stack pointer unchanged for a single push instruction fetched by said program control unit, as claim 10 recites.

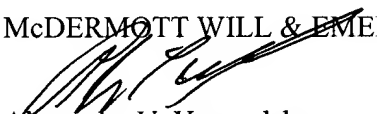
Accordingly, these claims are not obvious over Kudo et al. in view of Geldman et al. under 35 U.S.C. § 103(a).

Conclusion

For all of the foregoing reason, Appellant respectfully submits that the rejection of claims 1-10 under 35 U.S.C. § 103 is improper and should be reversed.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP



Alexander V. Yampolsky
Registration No. 36,324

Please recognize our Customer No. 20277 as our correspondence address.

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 SAB/AVY/dlb
Facsimile: 202.756.8087
Date: June 15, 2005

CLAIMS APPENDIX

1. (Original) A microprocessor including:

a program control unit controlling fetch of an instruction code;

an instruction decode unit decoding said fetched instruction code;

an address operation unit operating an address of a memory on the basis of the result of decoding by said instruction decode unit; and

a data operation unit operating data on the basis of the result of decoding by said instruction decode unit, wherein

said data operation unit executes data transfer between registers and data transfer between said registers and said memory in correspondence to single said instruction code having a single operation code fetched by said program control unit.

2. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to said memory and transfers data stored in a second register to said first register in correspondence to a single push instruction fetched by said program control unit.

3. (Original) The microprocessor according to claim 2, wherein said data operation unit decrements the value of a stack pointer after transferring said data stored in said second register to said first register.

4. (Original) The microprocessor according to claim 2, wherein said first register is a work register implemented in said data operation unit.

5. (Original) The microprocessor according to claim 2, wherein said second register is a control register implemented in one of said address operation unit and said program control unit.

6. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to a second register and transfers data stored in said memory to said first register in correspondence to a single pop instruction fetched by said program control unit.

7. (Original) The microprocessor according to claim 6, wherein said data operation unit increments the value of a stack pointer after transferring said data stored in said memory to said first register.

8. (Original) The microprocessor according to claim 6, wherein said first register is a work register implemented in said data operation unit.

9. (Original) The microprocessor according to claim 6, wherein said second register is a control register implemented in one of said address operation unit and said program control unit.

10. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to said memory and keeps the value of a stack pointer unchanged for a single push instruction fetched by said program control unit.